Antonia Jones

Reading 4 Response

The author of “API Design Matters”, Michi Henning, discusses how easy it is to make a bad API, and how much damage that can cause in the long run. He begins by giving an example of an API design that causes users a lot of grief just because of a few simple mistakes. He then goes on to discuss the drawbacks of poor APIs. Some import points he makes involves how APIs are designed to be used by other programmers. Designing an API poorly therefore goes on to affect many more programs than itself. Besides that if it is confusing it will probably cause others to misuse the code and therefore lead to more bugs and poor design down the line. Essentially, poor API design has a snowball effect where it can affect the efficiency, readability, and reliability of any program is used in. Henning then continues to discuss how API design can be used by emphasizing simplicity, clarity, and managing the details of the operation internally.

This is relevant to our Rails project in the sense that we are using the SaaS methodology. Because nothing is supposed to be shared between modules, APIs are essential to get code working together. It is important that the Controllers and Models for each component be able to interact with the others fluidly. This is especially important because we as developers cannot always work together in the same location, so clarity in our code is essential for things to work together correctly.

I agree with most of Henning’s statements. There are many common functions and libraries in languages such as C++ and Java that I use over and over because they are essential to most projects. When I find inconvenient quirks to those APIs it seems to bog down every project in which they appear. It is clear that a poor API design can damage countless other projects which must go on to use it and could therefore cause unknown amounts of damage in the long run.